CS1010S Programming Methodology

Lecture 12
The Last Lecture

15 April 2015

Practical Exam

- 18 Apr 2015 (Sat), 10am – 12nn
- COM1 PL 1, 2, 3 + I^3 WSL 1, 2, 3
- 3 x Questions + 1 x Bonus Question
- Open Book, No thumb drives

You need your IVLE login + password

Access to IDLE, IVLE, Coursemology + PythonTutor

You need your Coursemology login email + password

PE Questions

- General Problem Solving (Recursion/Iteration) Easy
- Data Processing (read + process data file) Ok
- OOP Minor challenges
- Bonus: mystery question So-so Challenging

Help Sessions

- Thursday, 16 April 6 – 8pm
- Tuesday, 21 April 6:30 – 8:30pm
- Thursday, 23 April 12 – 2pm

- Lots of trainings in Coursemology

Recitation

- Tomorrow's recitation will be review for Practical/Final
- Friday's recitation will be cancelled
  - Dr Ket Fah will be at SoC for consultation

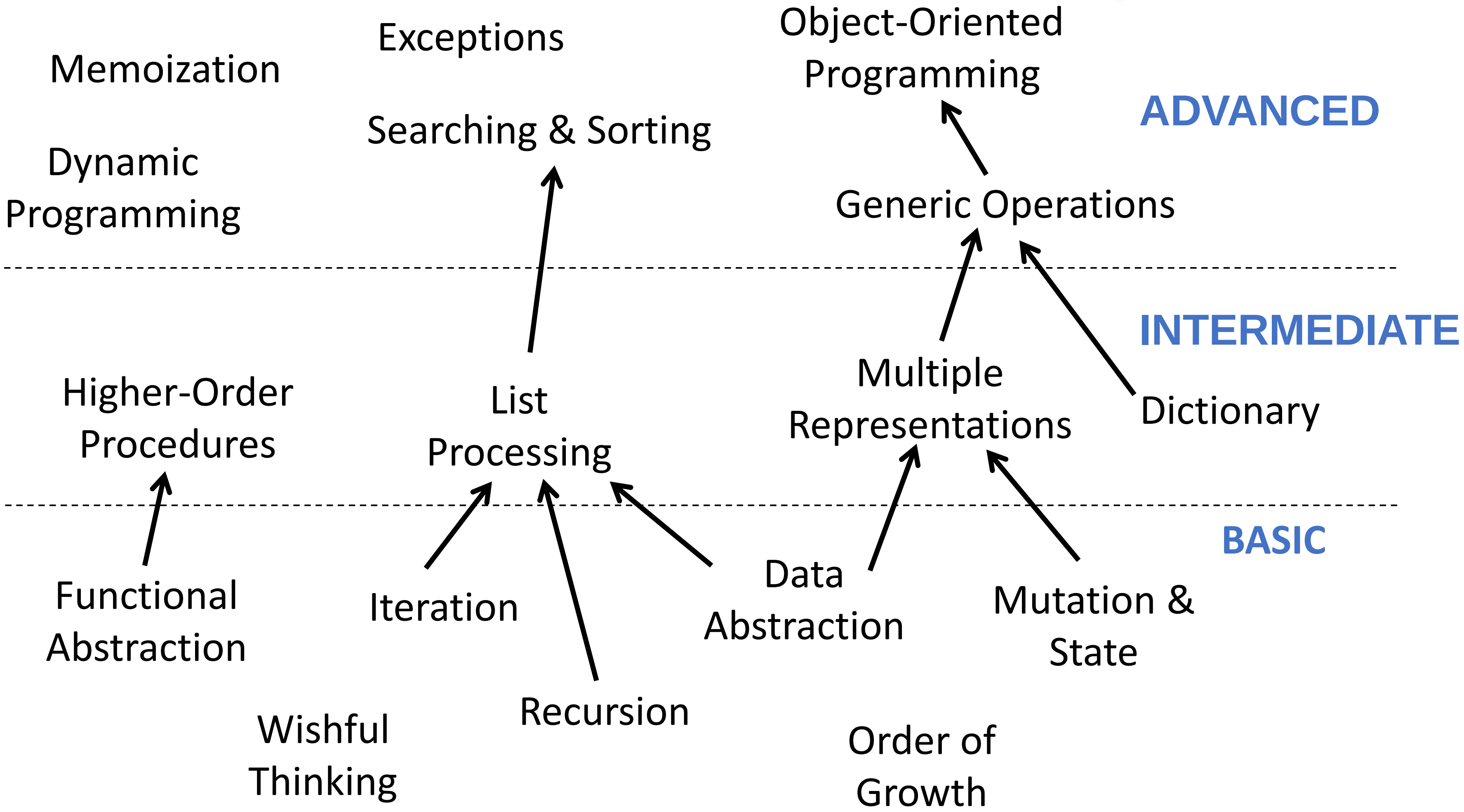
Final Exam

- Scope – Everything!
- 29 Apr (Wed) @5PM
- Open-Sheet
  - Can bring 2 x A4 sheets of notes (both sides)
- 2 hours, 100 marks total
- Manage time wisely
  - Do the easy questions first
  - Questions NOT in order of difficulty

What Did We Learn This Semester….

… what’s going to be on the exam

CS1010S Road Map



Memoization

Exceptions

Object-Oriented Programming

ADVANCED

Searching & Sorting

Dynamic Programming

Generic Operations

INTERMEDIATE

Higher-Order Procedures

List Processing

Multiple Representations

Dictionary

BASIC

Functional Abstraction

Iteration

Data Abstraction

Mutation & State

Wishful Thinking

Recursion

Order of Growth

Fundamental <u>concepts</u> of computer programming

Week 1

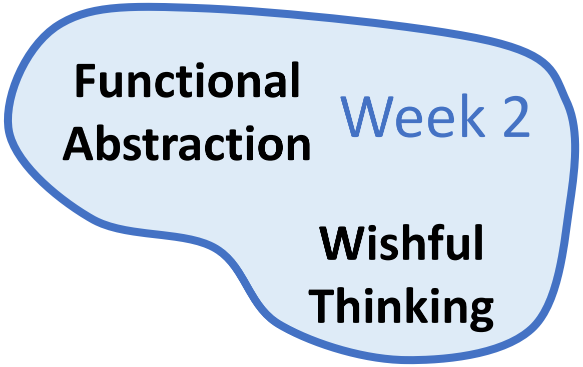
Syntax + Conditionals

CS1010S Road Map

ADVANCED

INTERMEDIATE

BASIC



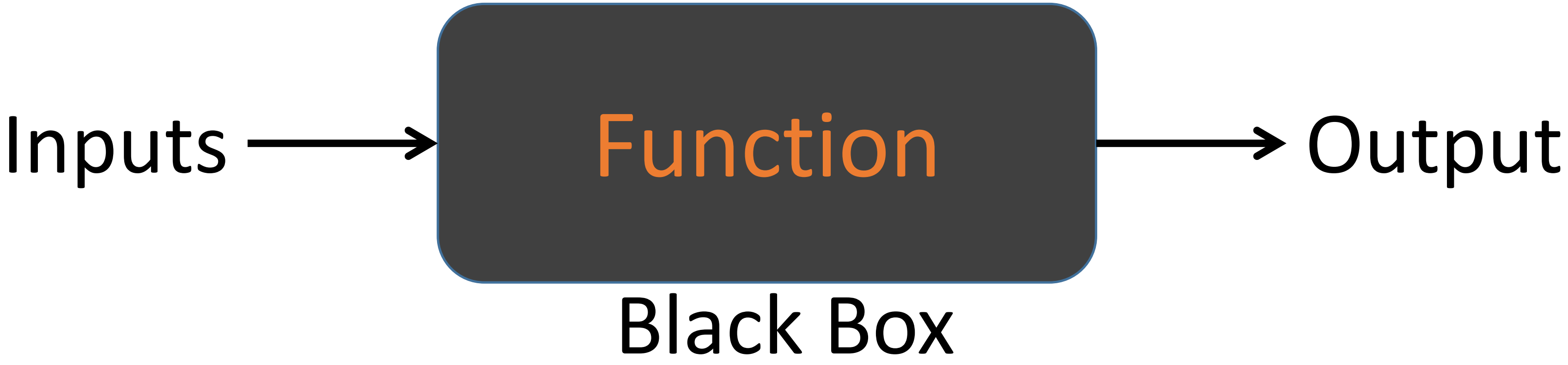
Functional Abstraction

Week 2

Wishful Thinking

Fundamental **concepts** of computer programming

Functional Abstraction



`lambda`

create functions

`def fn(…):`

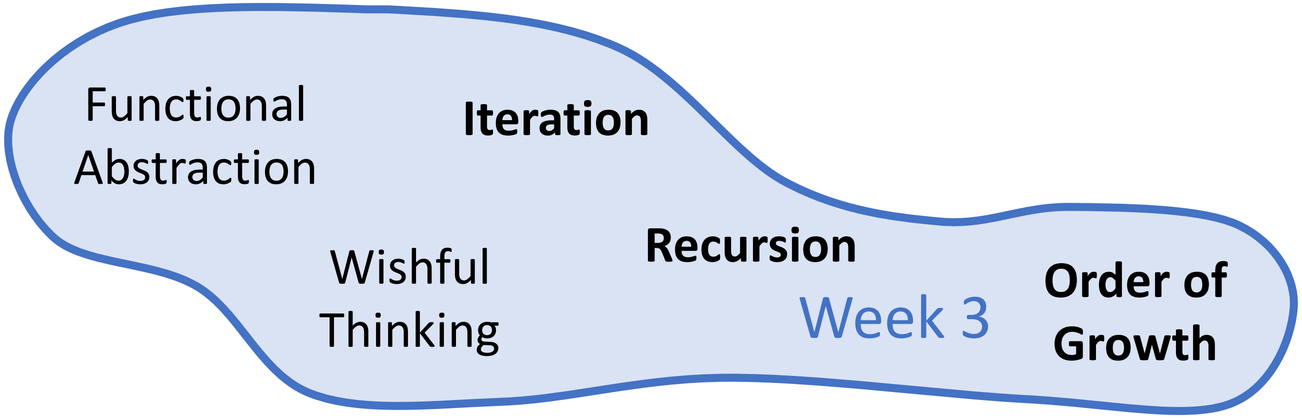
lambda + variable binding

CS1010S Road Map

ADVANCED

INTERMEDIATE

BASIC



Functional Abstraction

Iteration

Wishful Thinking

Recursion

Week 3

Order of Growth

Fundamental <u>concepts</u> of computer programming

Factorial

$$n! = \begin{cases} n \times (n-1)!\,, & n > 1 \\ 1\,, & n = 1 \end{cases}$$

```
def factorial(n):
    if n <= 1:
        return 1
    else:
        return n * factorial(n – 1)
```

Recursion

1. Base Case
2. Recursive Step

Iterative Factorial

$$n! = 1 \times 2 \times 3 \cdots \times n$$

Factorial rule:

product ← product × counter

counter ← counter + 1

```
def factorial(n):
    product = 1
    for counter in range(2, n+1):
        product = product * counter
    return product
```

Iteration

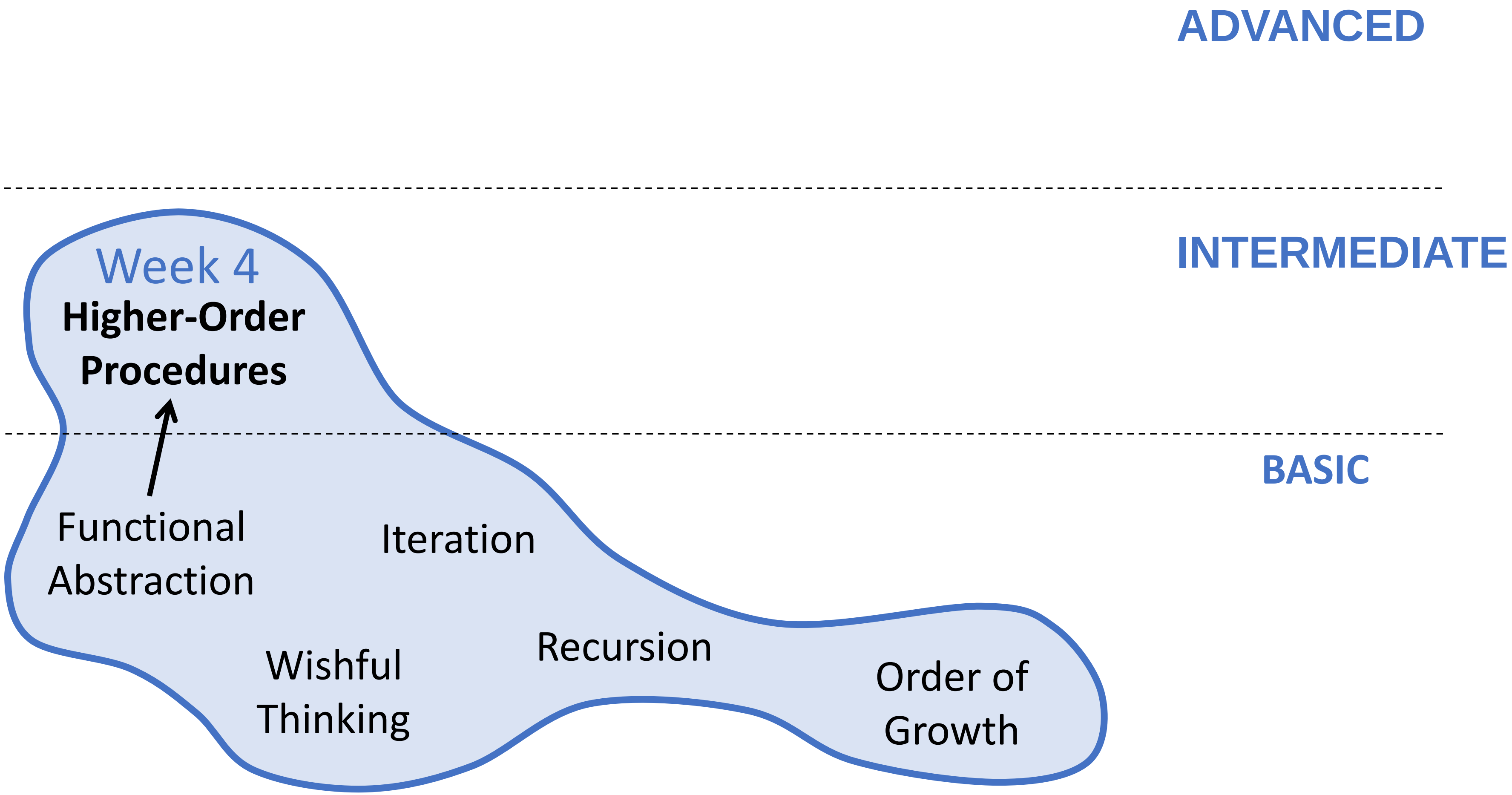
for loop

```
for i in range(start, stop, step):
    # do stuff
for item in seq:
    # do stuff
```

while loop

```
while cond:
    # update cond
```

CS1010S Road Map



ADVANCED

INTERMEDIATE

BASIC

Week 4

Higher-Order Procedures

Functional Abstraction

Iteration

Wishful Thinking

Recursion

Order of Growth

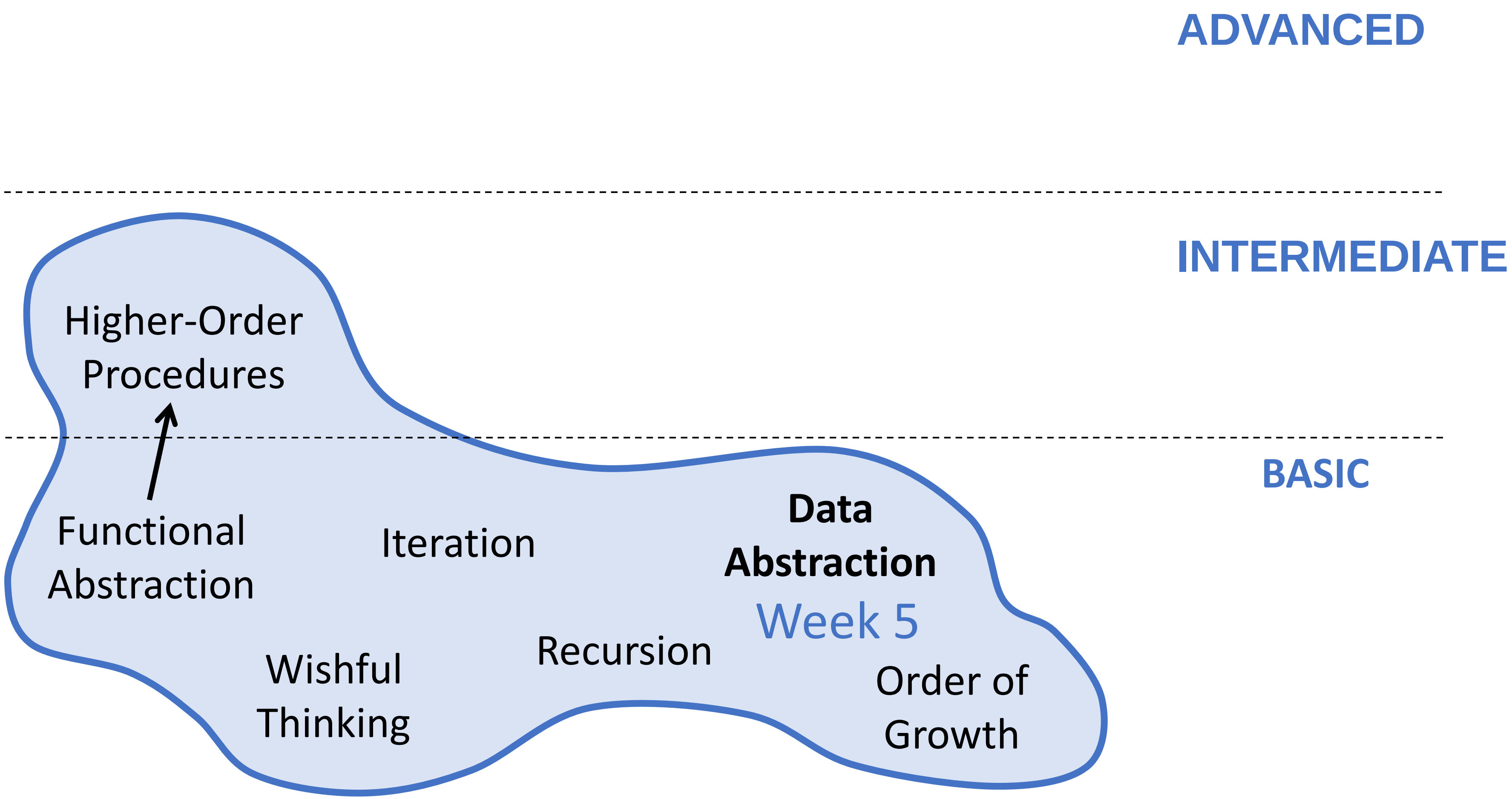
Fundamental <u>concepts</u> of computer programming

Higher-Order Functions

- Functions can be inputs to functions
- Functions can be return values from functions

- Capturing common patterns

CS1010S Road Map



ADVANCED

INTERMEDIATE

BASIC

Higher-Order Procedures

Functional Abstraction

Iteration

Data Abstraction

Week 5

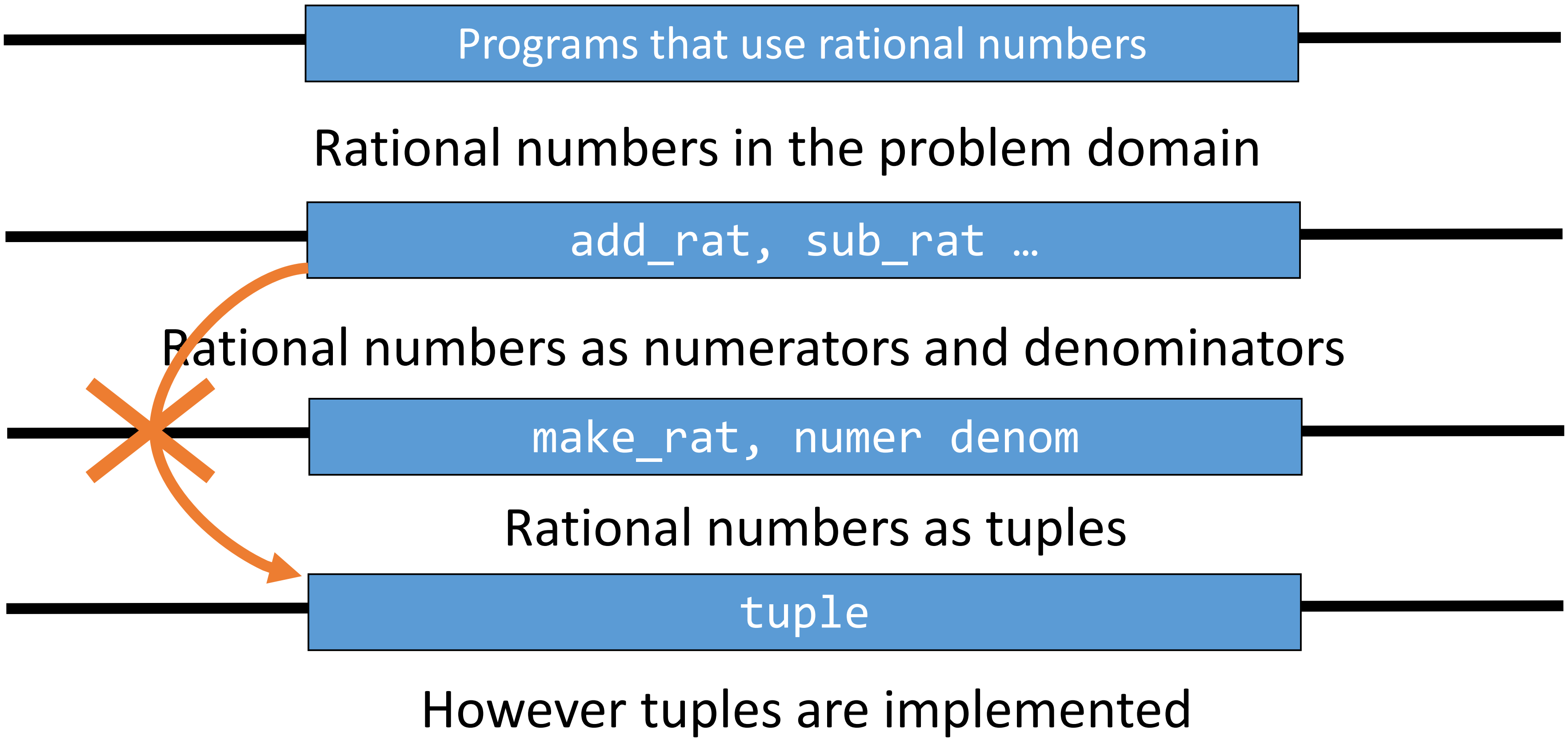
Wishful Thinking

Recursion

Order of Growth

Fundamental **concepts** **of computer programming**

Abstraction Barrier



Programs that use rational numbers

Rational numbers in the problem domain

add_rat, sub_rat …

Rational numbers as numerators and denominators

make_rat, numer denom

Rational numbers as tuples

tuple

However tuples are implemented

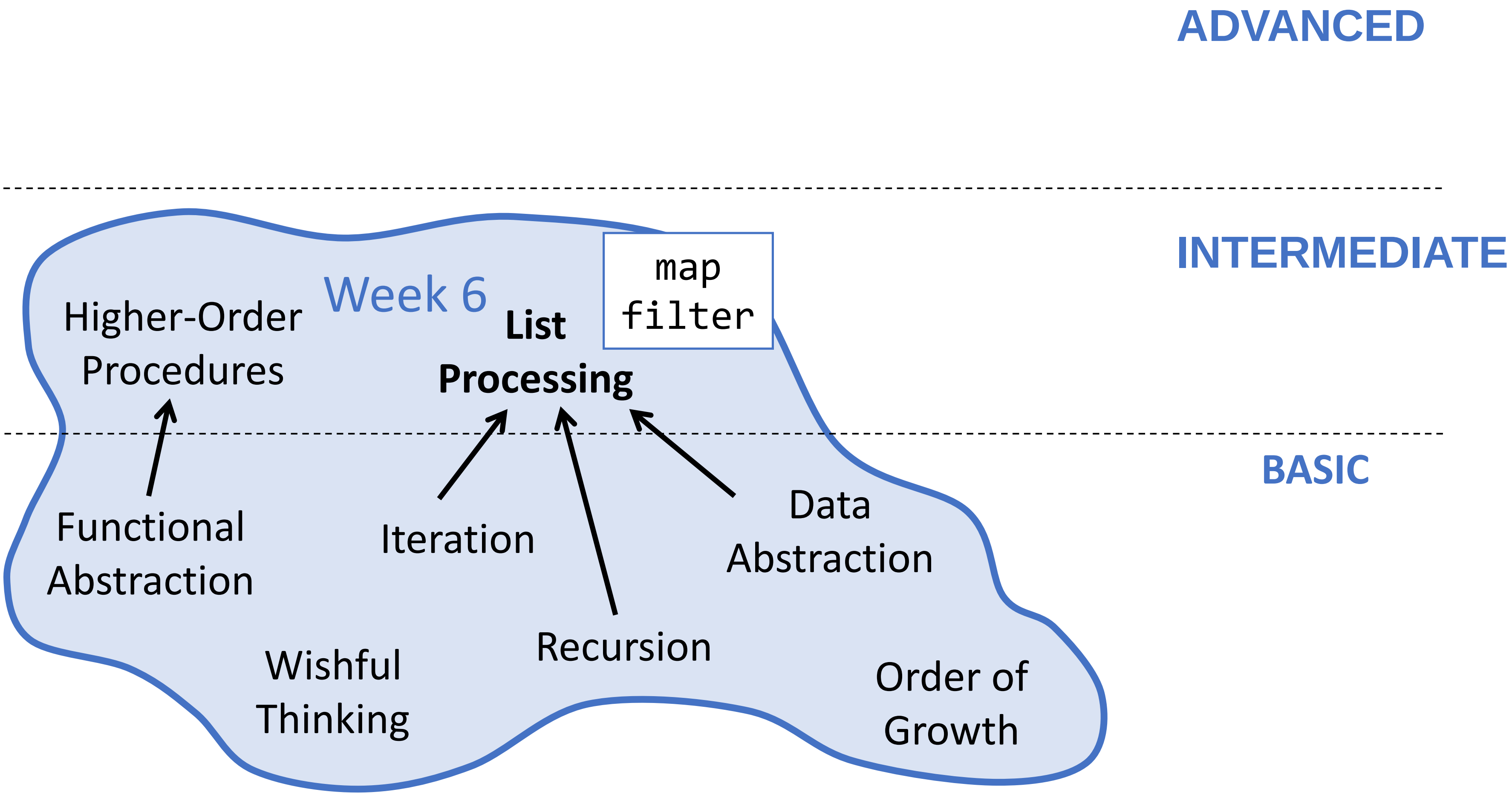
At each level, use only functions available at that interface, not below it.

Concepts of Equality:

Equivalence (==)

Identity (is)

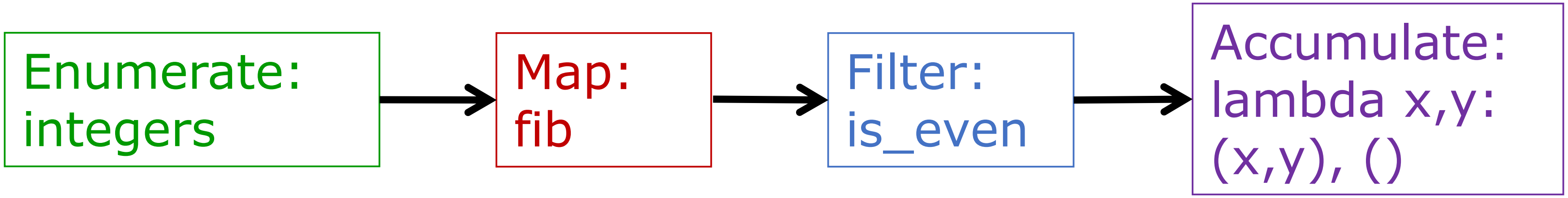
CS1010S Road Map



Fundamental **concepts** of computer programming

List and Sequences

- `map` and `filter`
- Signal processing view
  - `even_fibs`



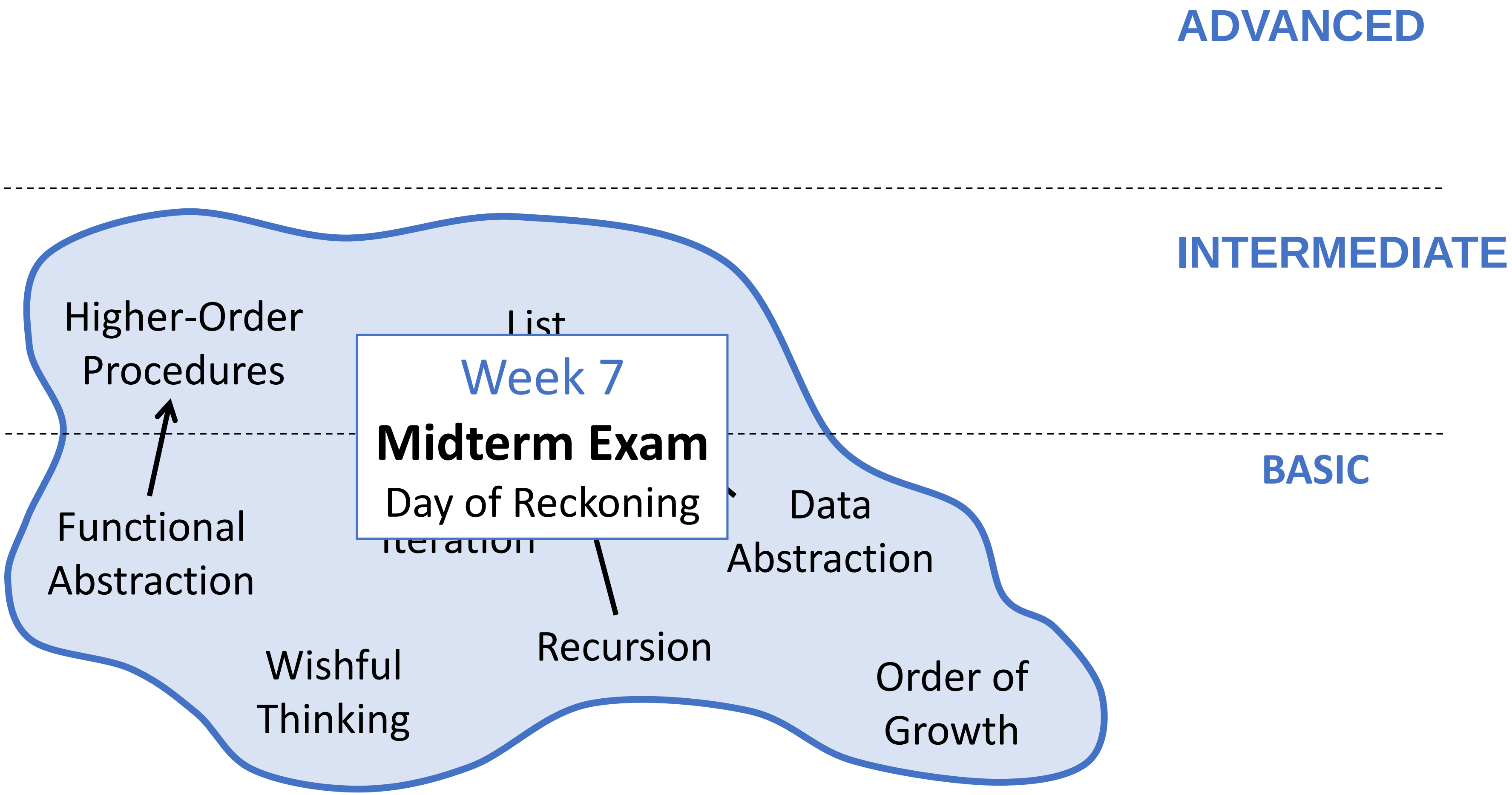
Enumerate: integers

Map: fib

Filter: is_even

Accumulate: lambda x,y: (x,y), ()

CS1010S Road Map



ADVANCED

INTERMEDIATE

BASIC

Higher-Order Procedures

List

Functional Abstraction

Iteration

Data Abstraction

Wishful Thinking

Recursion

Order of Growth

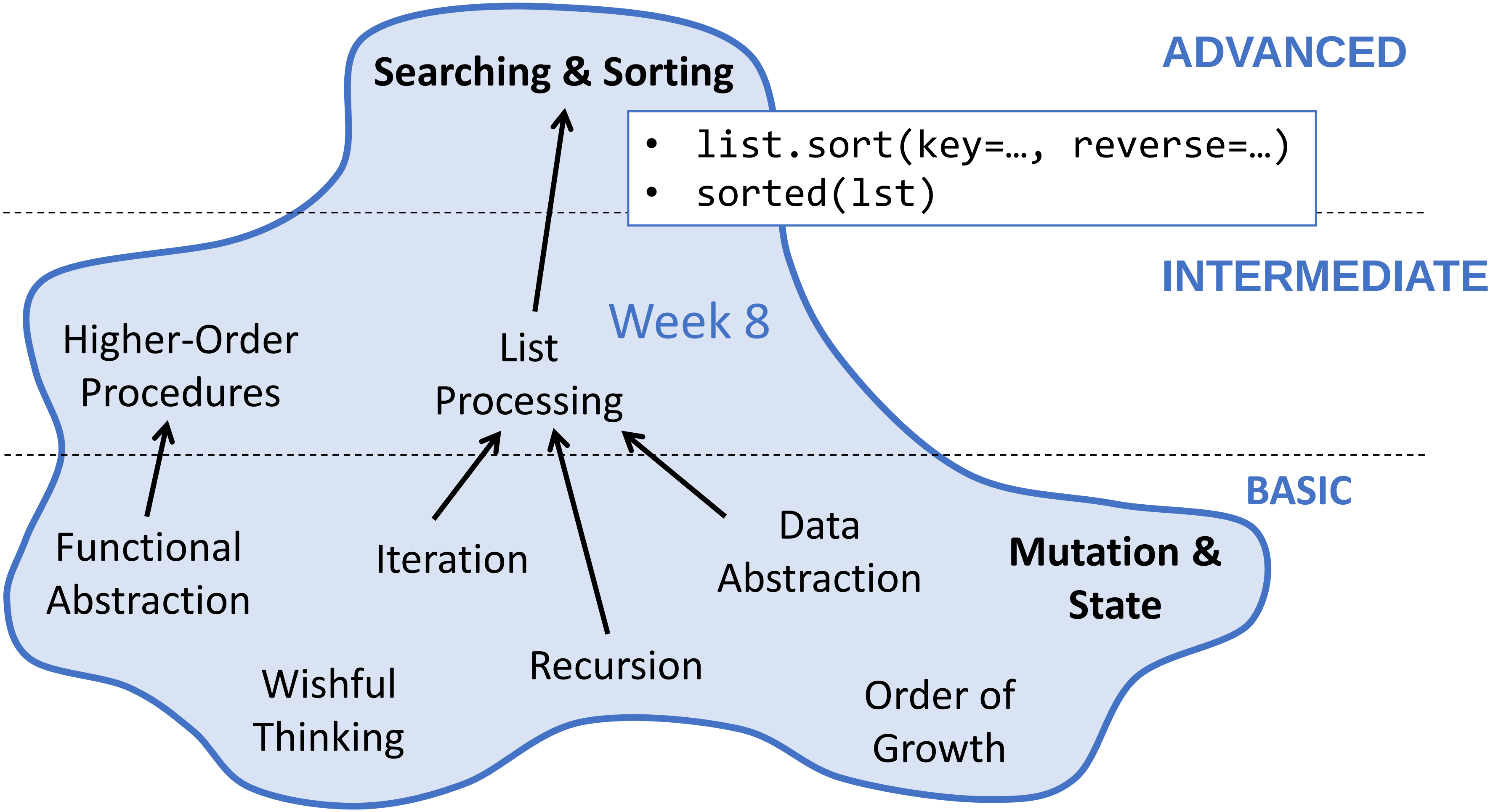
Week 7

Midterm Exam

Day of Reckoning

Fundamental <u>concepts</u> of computer programming

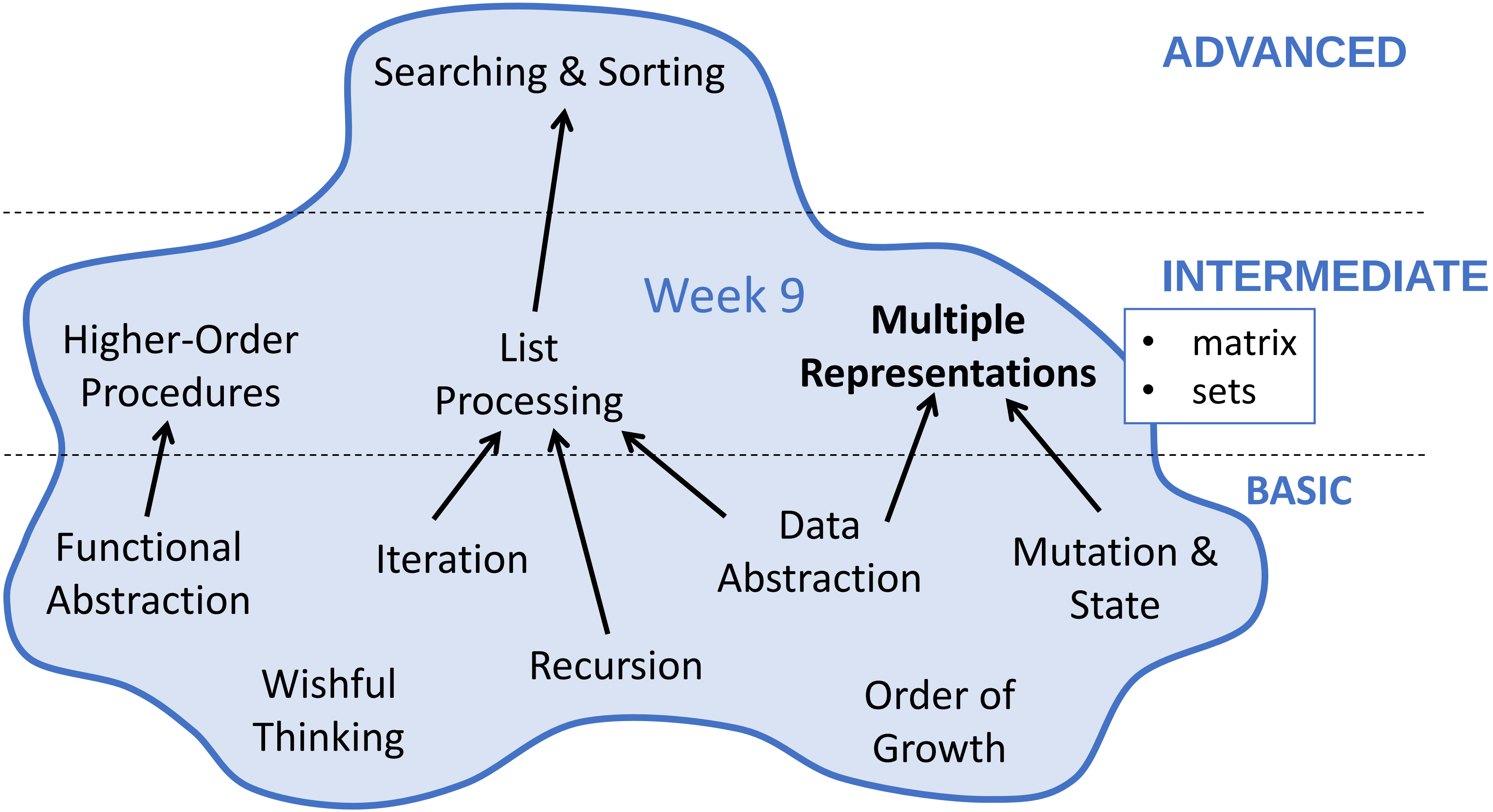
CS1010S Road Map



- `list.sort(key=…, reverse=…)`
- `sorted(lst)`

Fundamental **concepts** of computer programming

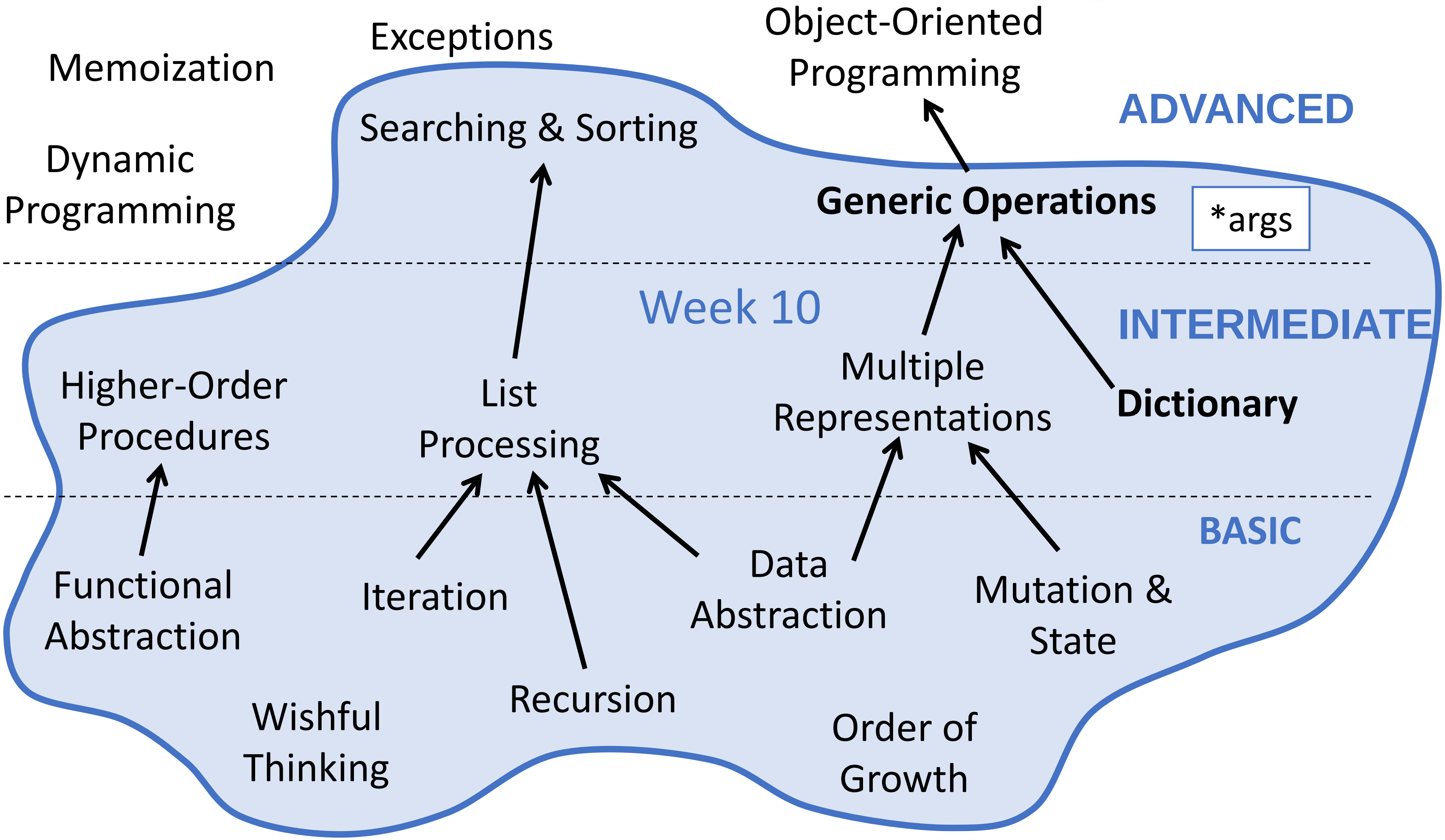
CS1010S Road Map



- matrix
- sets

Fundamental concepts of computer programming

CS1010S Road Map



Memoization

Exceptions

Object-Oriented Programming

ADVANCED

Searching & Sorting

Dynamic Programming

Generic Operations

*args

Week 10

INTERMEDIATE

Higher-Order Procedures

List Processing

Multiple Representations

Dictionary

BASIC

Functional Abstraction

Iteration

Data Abstraction

Mutation & State

Recursion

Wishful Thinking

Order of Growth

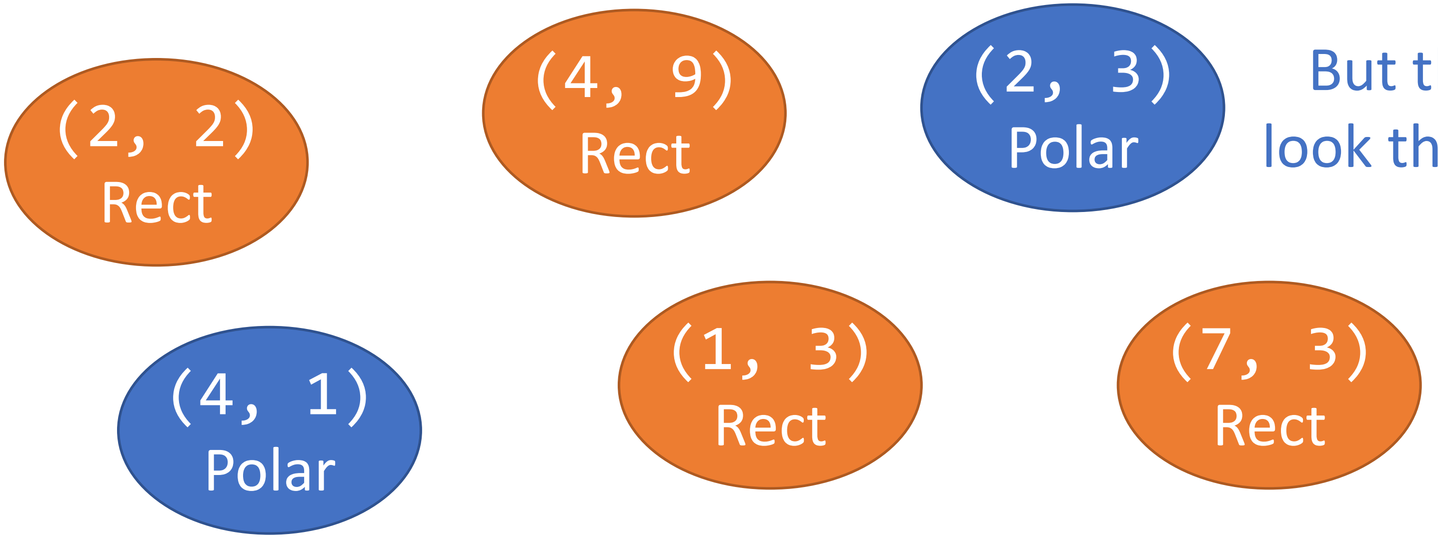
Fundamental **concepts** **of computer programming**

Rect version:

```
make_from_real_imag(x, y)
real_part(z)
imag_part(z)
magnitude(z)
angle(z)
make_from_mag_ang(r, a)
```

Polar version:

```
make_from_real_imag(x, y)
real_part(z)
imag_part(z)
magnitude(z)
angle(z)
make_from_mag_ang(r, a)
```



(2, 2)
Rect

(4, 9)
Rect

(2, 3)
Polar

(4, 1)
Polar

(1, 3)
Rect

(7, 3)
Rect

But they all
look the same!

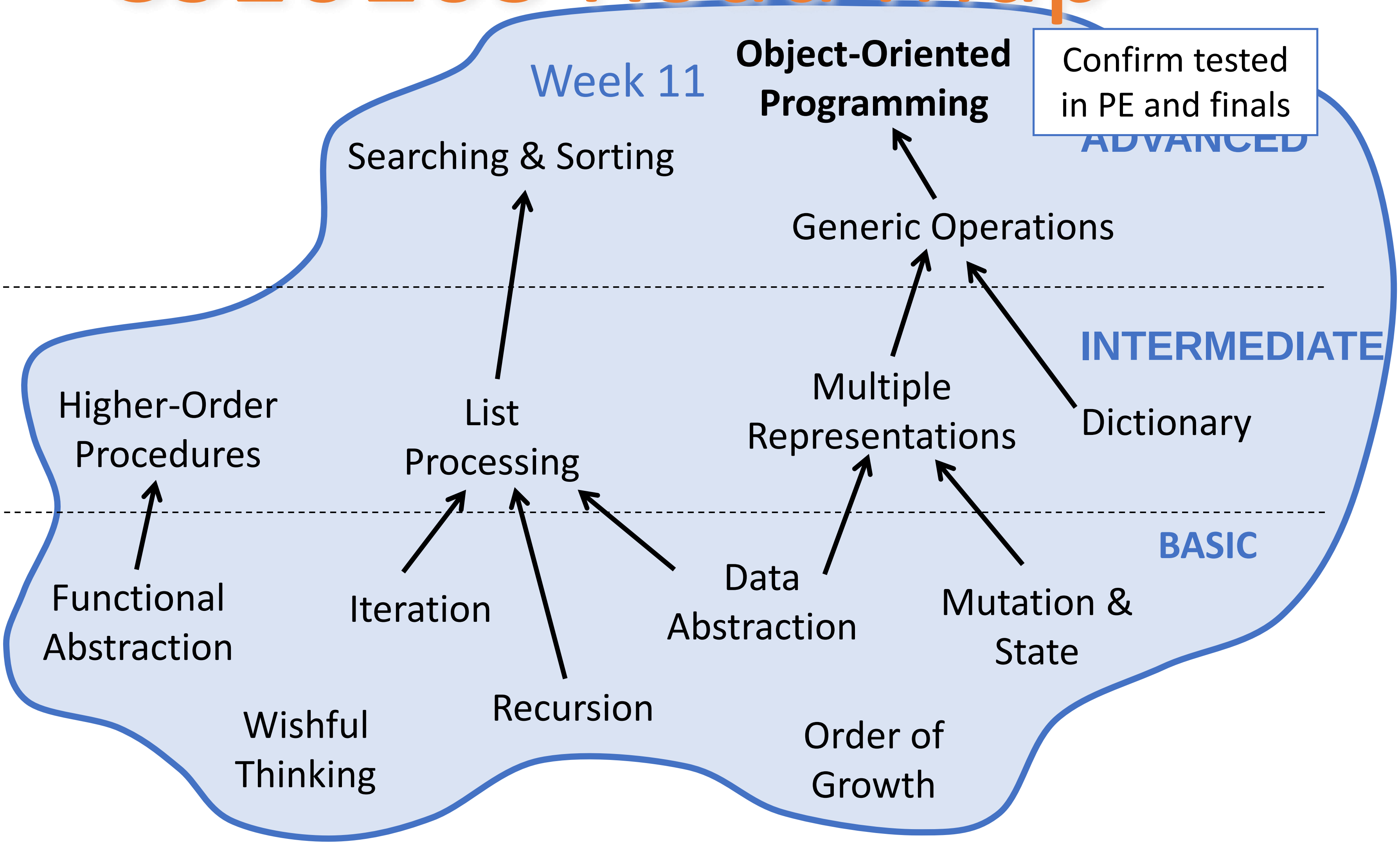
All tuples!!

3 Strategies

- Dispatch on Type
  - `if-else`
- Data Directed Programming
  - Store in table
- Message Passing
  - Put function in the data!

Won't be tested in PE or Finals

CS1010S Road Map



Week 11

Object-Oriented Programming

Confirm tested in PE and finals

ADVANCED

Searching & Sorting

Generic Operations

INTERMEDIATE

Higher-Order Procedures

List Processing

Multiple Representations

Dictionary

BASIC

Functional Abstraction

Iteration

Data Abstraction

Mutation & State

Wishful Thinking

Recursion

Order of Growth

Fundamental **<u>concepts</u>** **of computer programming**

OOP =

Message Passing +

Data Abstraction

Major concepts

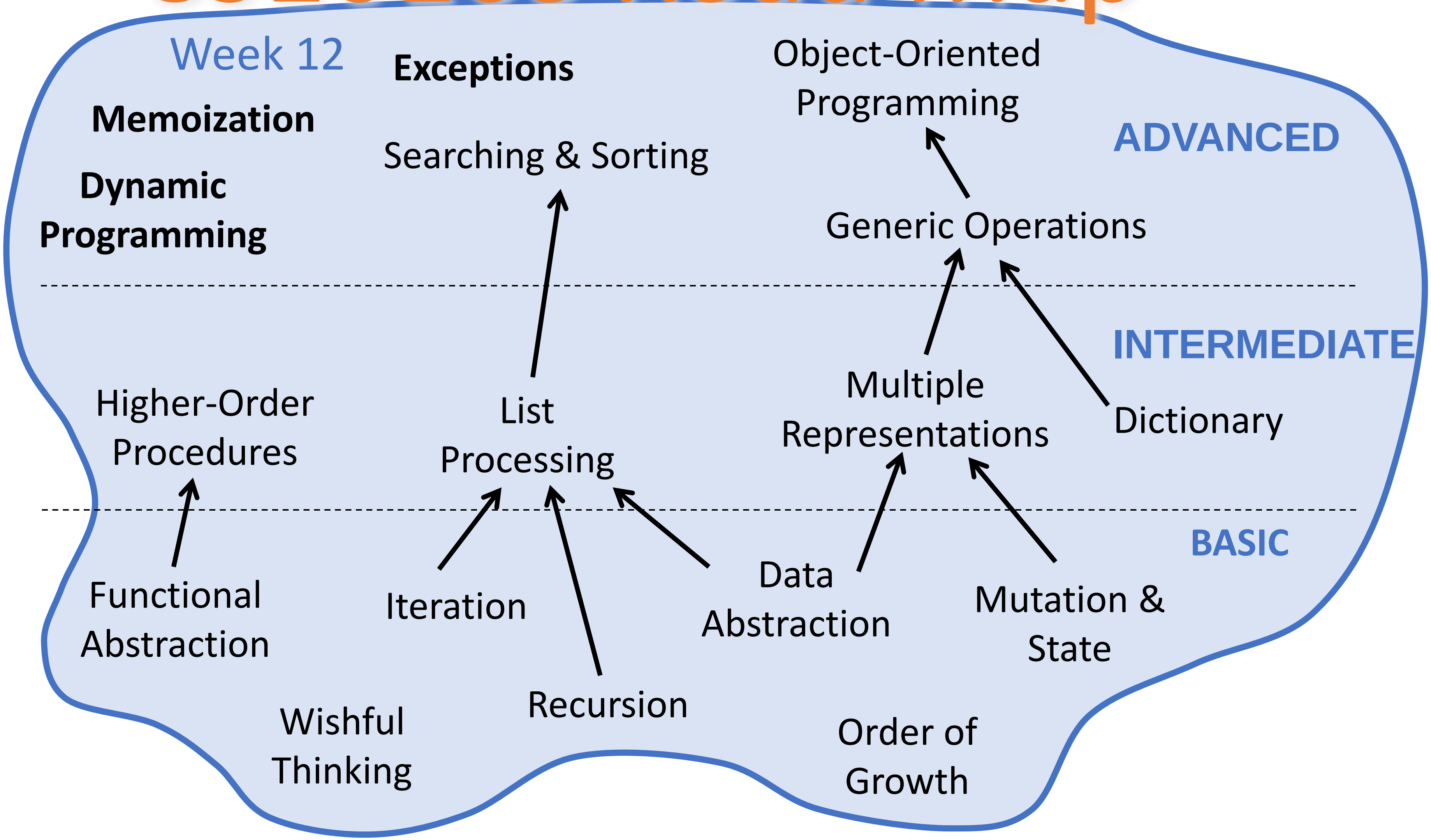
- Classes and instances
- Properties (state)
- Methods
- Inheritance
- Polymorphism (overriding)

Focus on syntax

`self`

`super()`

CS1010S Road Map



Week 12

Exceptions

Memoization

Dynamic Programming

Searching & Sorting

Object-Oriented Programming

ADVANCED

Generic Operations

INTERMEDIATE

Higher-Order Procedures

List Processing

Multiple Representations

Dictionary

BASIC

Functional Abstraction

Iteration

Data Abstraction

Mutation & State

Wishful Thinking

Recursion

Order of Growth

Fundamental <u>concepts</u> of computer programming

Won’t come out on PE/Finals

Memoization

Redundant Computations!

Dynamic Programming

remember what you computed before in table!

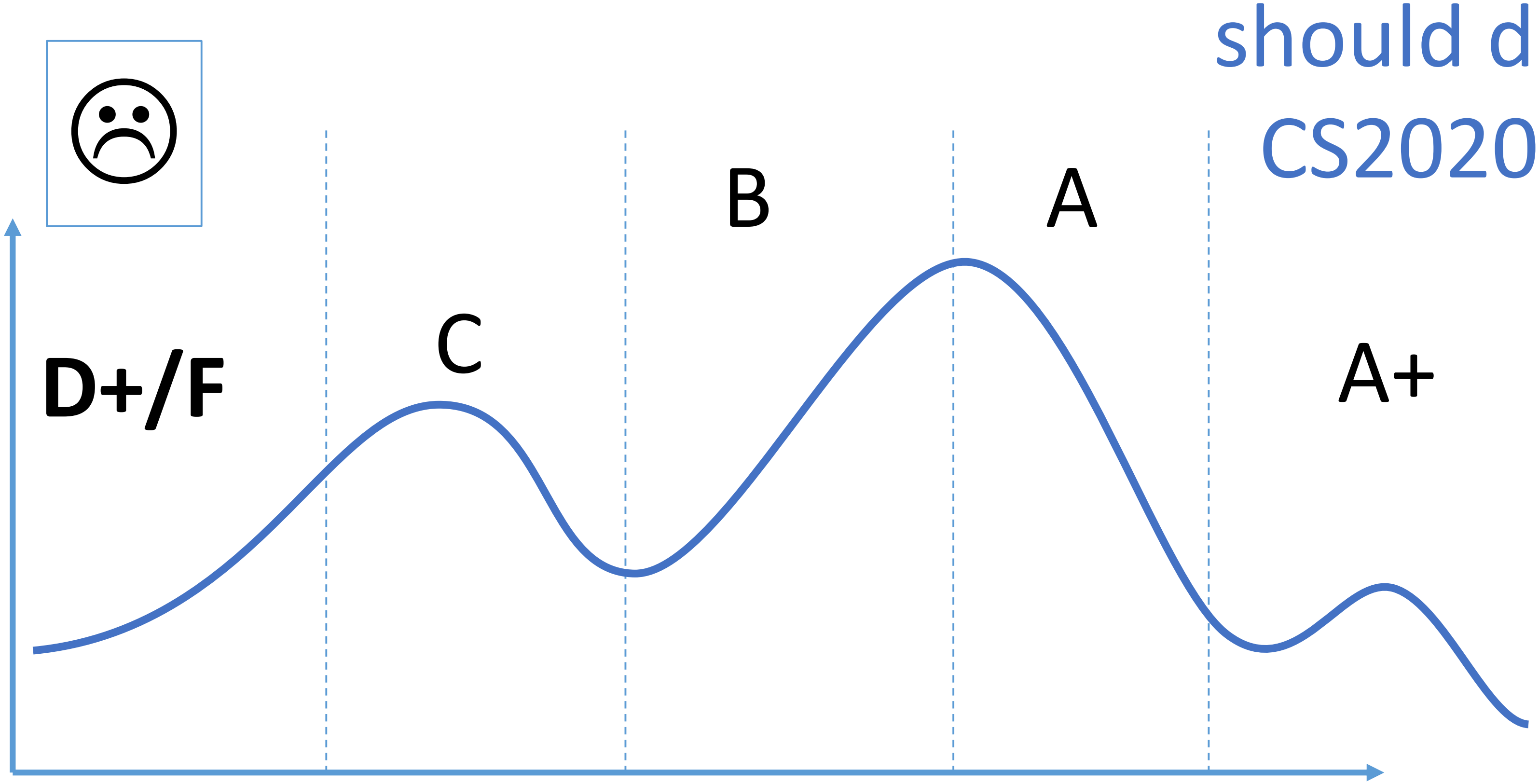
fill the table directly!

3 Types of Data Structures

• Tuple ( )	immutable indexed by int
• List [ ]	mutable indexed by int
• Dictionary { }	mutable indexed by key

How You Will Be Graded

should do CS2020



☹

D+/F

C

B

A

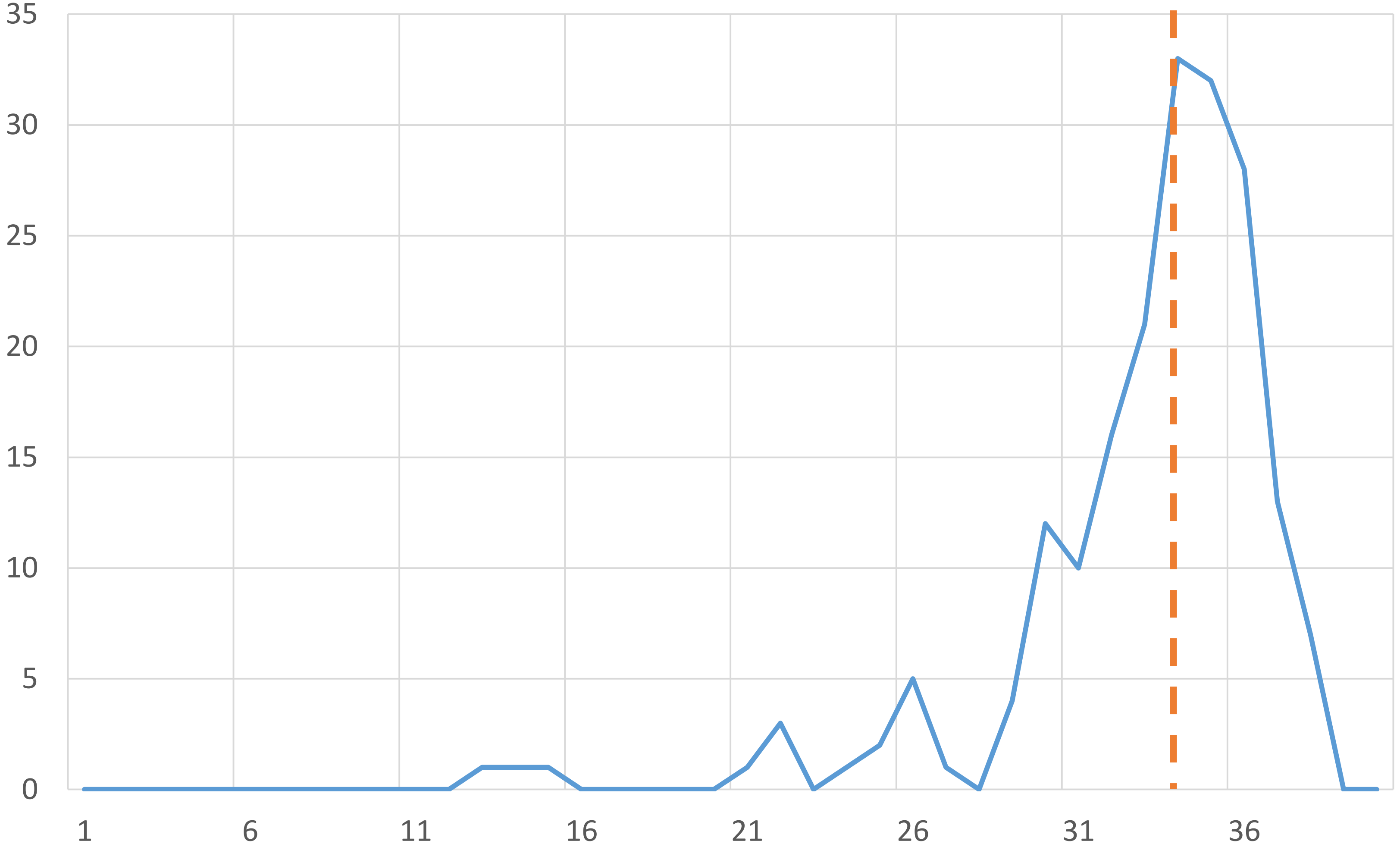
A+

solve basic problems, behind in homework

competent, cannot solve 25% of questions

can solve most questions, minor mistakes, Lvl 35

off the charts, Lvl 36++


35
30
25
20
15
10
5
0
1
6
11
16
21
26
31
36

5 years after you graduate
(9 years from now), when people ask
you what you learnt in college….

MANAGING COMPLEXITY

This is It